



Московский институт электроники и
математики имени А. Н. Тихонова

Кафедра информационной
безопасности киберфизических
систем

Москва 2025

Лекция 1: Введение в наступательную безопасность

Курс: Технологии пентестинга

Автор: Космачев Алексей Алексеевич



[~]\$ whoami

- Космачев Алексей Алексеевич
- > 4 лет опыта в ИБ
- 2 степени Бакалавра: НИУ ВШЭ МИЭМ “Информационная безопасность” и ФКН “Прикладная математика и информатика”
- Master of Information Security (Университет Иннополис)
- Ведущий пентестер в Bi.Zone, Ex. CERT-аналитик, Back-end разработчик
- Преподаватель в НИУ ВШЭ (Технологии пентестинга, дипломные работы, учебные практики)
- Победитель VI Кубка CTF России
- BSCP, OSWE
- Автор BDU, CVE
- Kaspersky Security Researchers Hall of Fame





Содержание дисциплины

1. Введение в наступательную безопасность, термины и определения
2. Разведка (OSINT)
3. Основы работы веб-приложений
- 4. Серверные уязвимости веб-приложений**
- 5. Клиентские уязвимости веб-приложений**
6. Реверс-инжиниринг и бинарная эксплуатация
7. Пост-эксплуатация в Linux (Pivoting, повышение привилегий, закрепление)





Элементы контроля

1. Практические занятия (Опр)
2. Контрольная работа (Ок)
3. Экзамен (Оэ)

Формула оценки: $0.5 * \text{Опр} + 0.2 * \text{Ок} + 0.3 * \text{Оэ}$



План лекции

1. Положение вещей
2. Основные понятия
3. Виды работ
4. Этапы проведения тестирования на проникновение
5. Методологии проведения тестирования на проникновение



Положение вещей

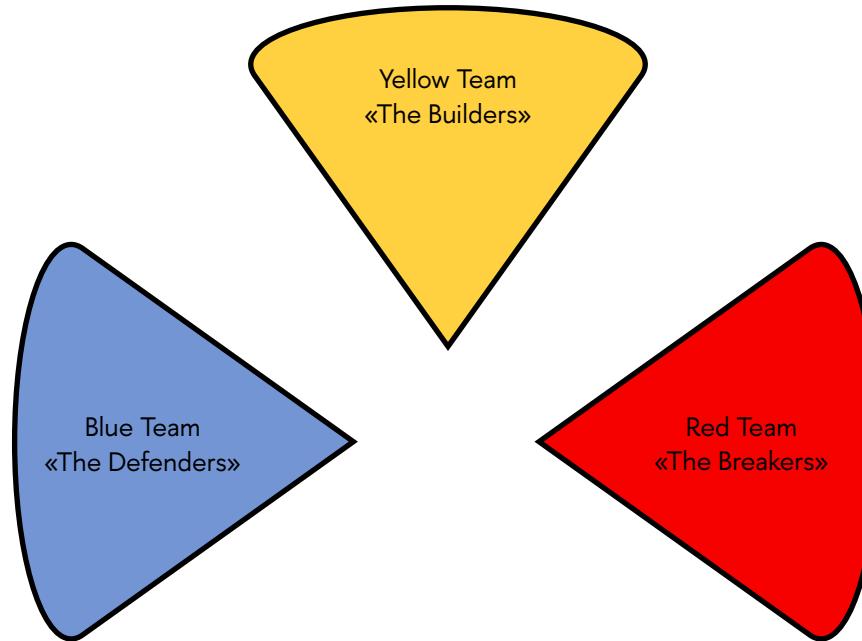


«В этом мире нет ничего, что невозможно взломать»*

*Все зависит от навыков, команды, потраченного времени и имеющихся технологий



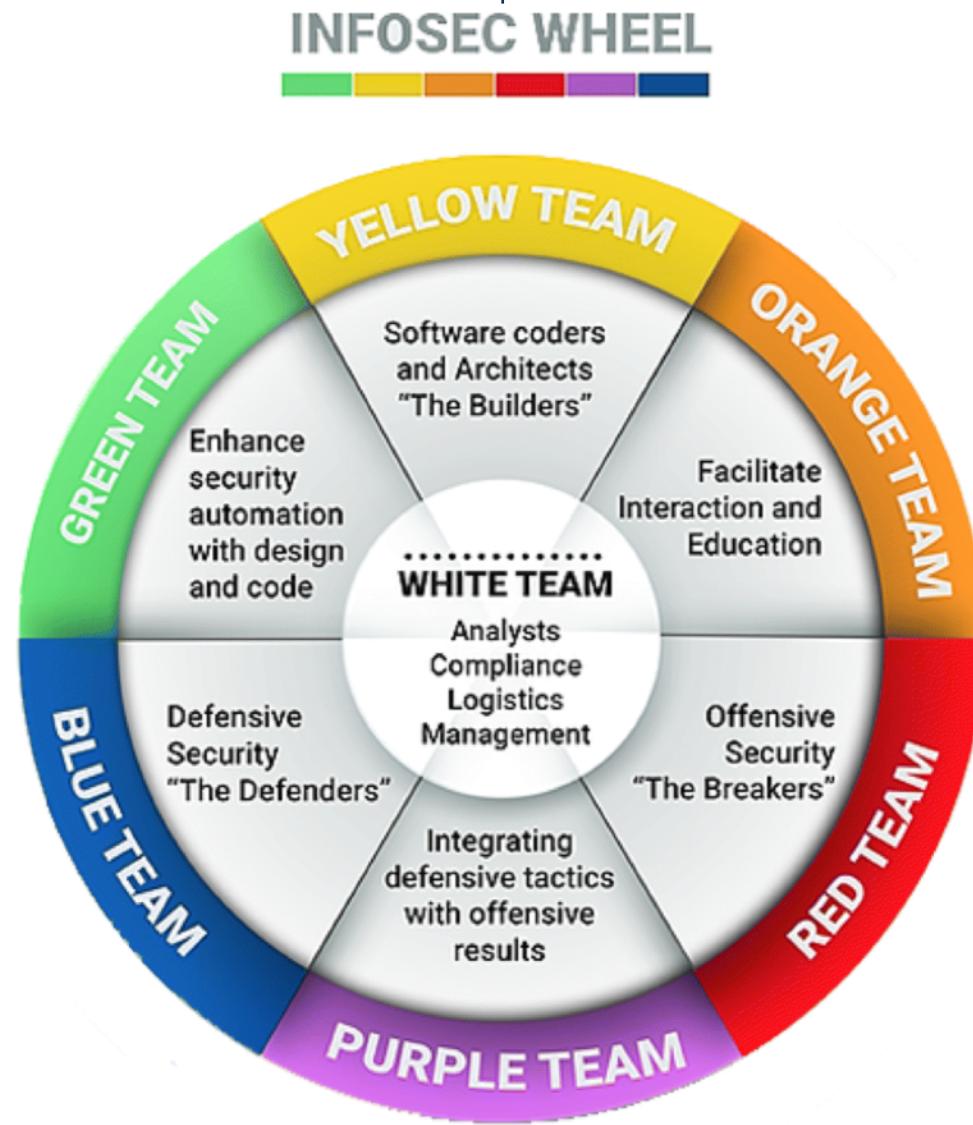
«Цветовое» разделение команд информационной безопасности



Основные направления информационной безопасности



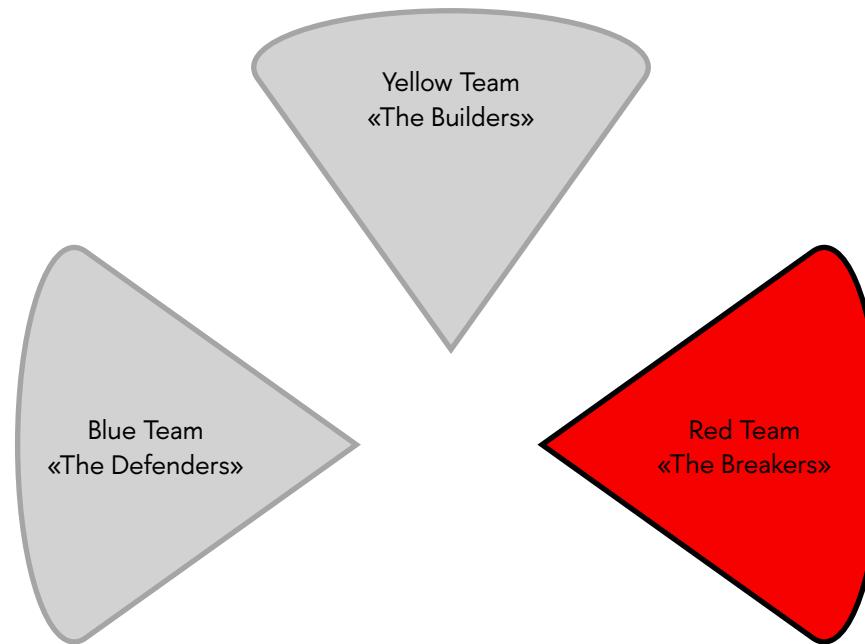
На самом деле все немного сложнее...



https://www.researchgate.net/publication/368786961_Cybercompetitions_A_survey_of_competitions_tools_and_systems_to_support_cybersecurity_education?tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Il9kaXJY3QiLCJwYWdlijoiX2RpcmVjdCJ9fQ



«Цветовое» разделение команд информационной безопасности

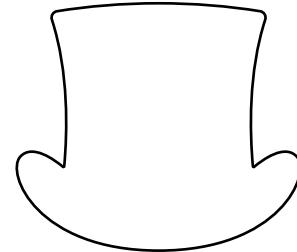


Основные направления информационной безопасности

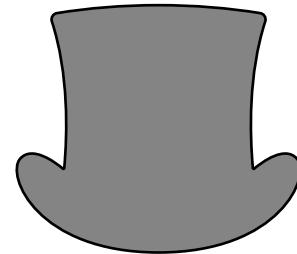


Хакеры и их разновидности

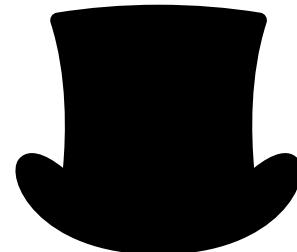
- «**Хакер**» ранее - человек, в подробностях разбирающийся в сложных системах, программист, способный быстро и элегантно решить трудную задачу, «Гик».
- «**Хакер**» сейчас - компьютерный взломщик, проникающий в закрытые информационные сети, банки данных и т. п. с целью получения доступа к секретной информации, а также заражения их вирусом. (Oxford Languages).



«Белые шляпы» - этичные хакеры.
Мотивация: сделать систему более
защищенной



«Серые шляпы»
Мотивация: деньги



«Черные шляпы» - преступники.
Мотивация: нанесение ущерба

Разновидности «шляп» хакеров



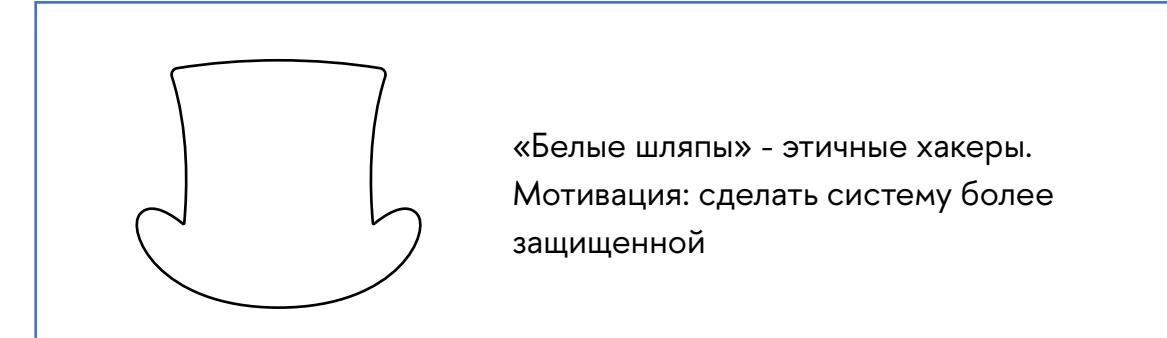
В реальности все снова
немного сложнее...



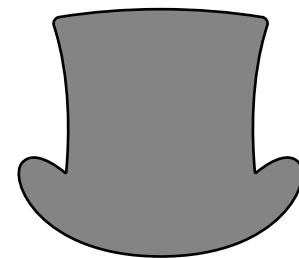


Хакеры и их разновидности

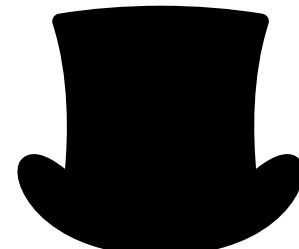
- «**Хакер**» ранее - человек, в подробностях разбирающийся в сложных системах, программист, способный быстро и элегантно решить трудную задачу, «Гик».
- «**Хакер**» сейчас - компьютерный взломщик, проникающий в закрытые информационные сети, банки данных и т. п. с целью получения доступа к секретной информации, а также заражения их вирусом. (Oxford Languages).



«Белые шляпы» - этичные хакеры.
Мотивация: сделать систему более
защищенной



«Серые шляпы»
Мотивация: деньги



«Черные шляпы» - преступники.
Мотивация: нанесение ущерба

Разновидности «шляп» хакеров



Основные понятия



Риск

Угроза



Уязвимость

Баг



CVE

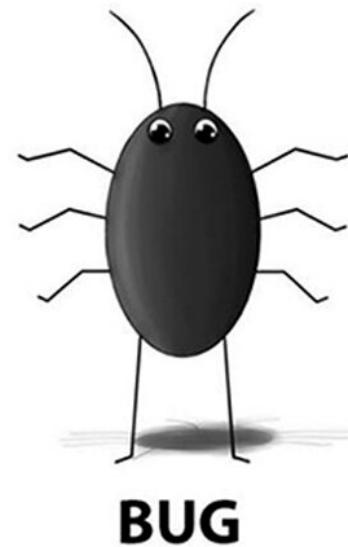
CWE



Баг

- Дефект; любая проблема, аномалия, ошибка в программном продукте, ведущая к некорректной работе отдельных компонентов или всему продукту в целом
- Получил свое название буквально из-за насекомого

Пример: «поплывшая» разметка на странице





Уязвимость

- **Баг**, ведущий к нанесению системе ущерба любого рода («злой» баг)
- Свойство информационной системы, предоставляющее возможность реализации **угроз** безопасности обрабатываемой в ней информации.
- В контексте информационной безопасности зачастую уязвимость = баг

Пример: SQL-инъекция в функционале поиска новостей в ленте

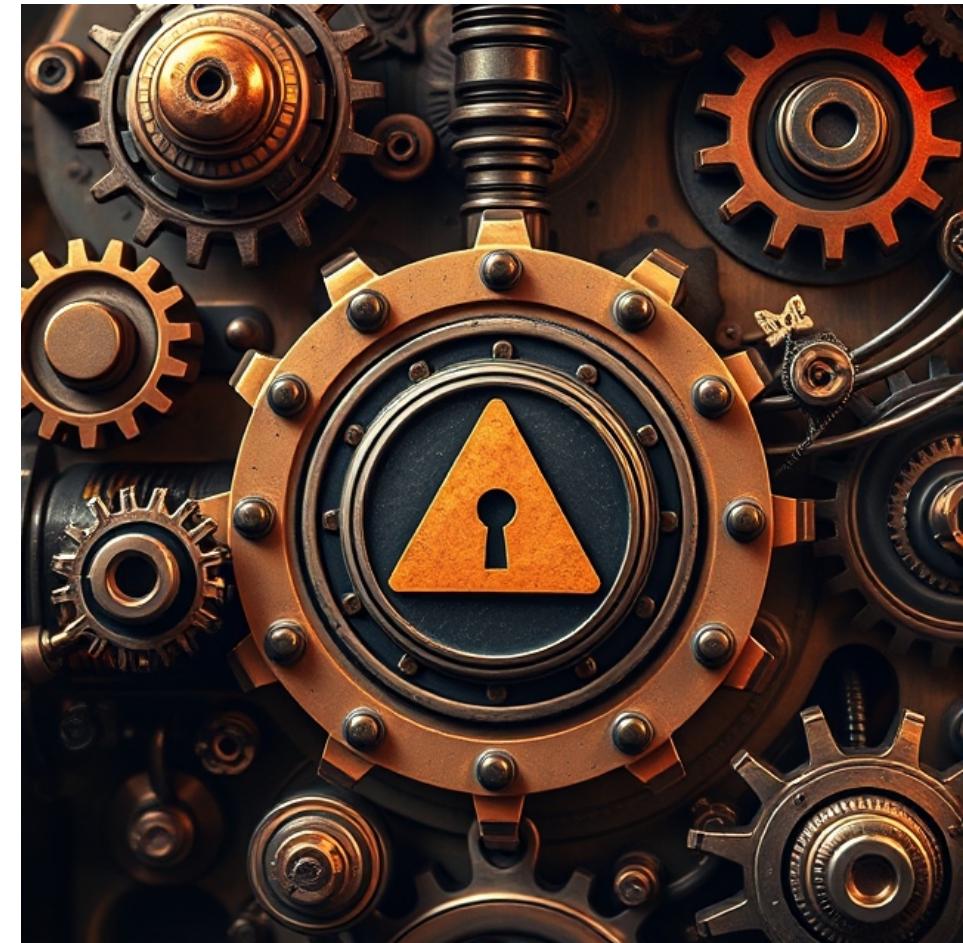




Угроза

- Совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации
- Что-то плохое, что может произойти
- Реализуется при эксплуатации **уязвимости**
- Описывает негативный эффект **риска**

Пример: Кража учетных данных пользователей





Риск

- Потенциальная возможность того, что **уязвимость** будет использоваться для создания **угрозы** активу или группе активов, приводящей к ущербу для организации
- **Угроза + Уязвимость (+ Актив)**



+

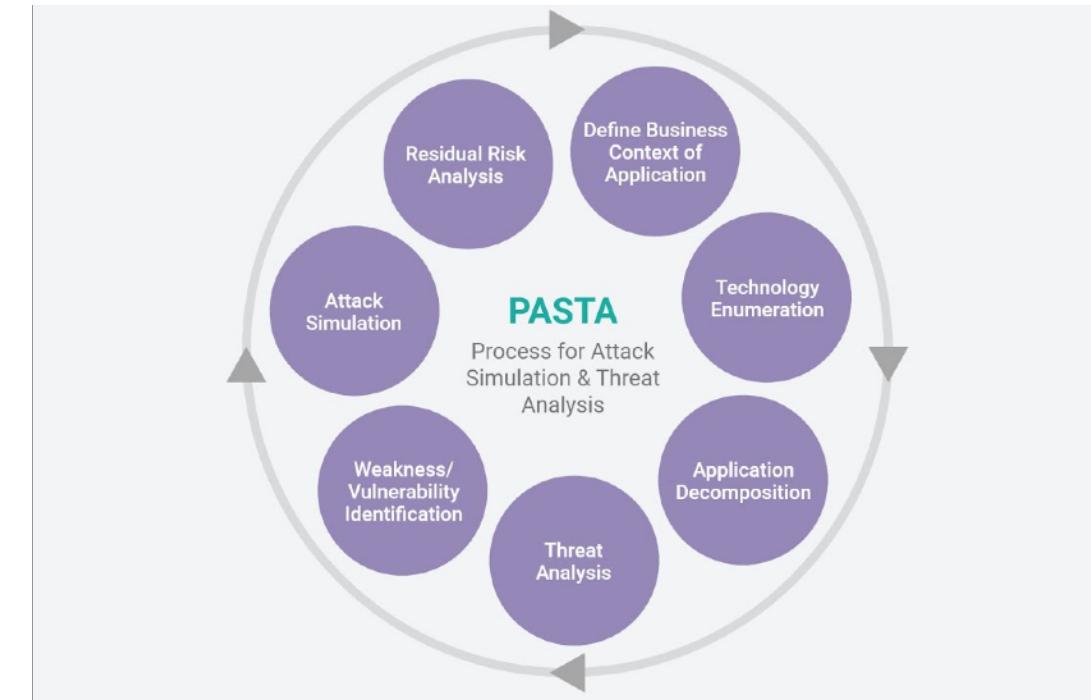
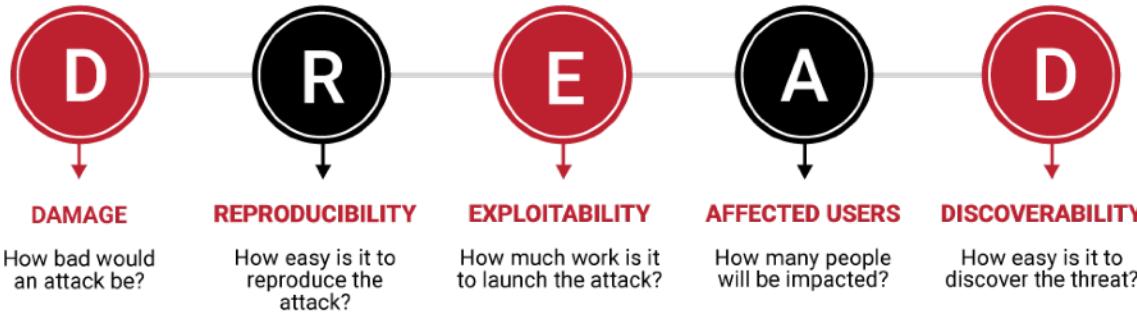
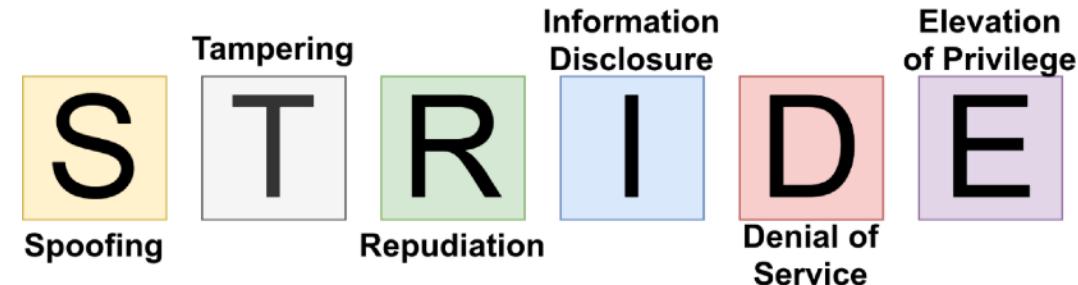


Пример: Кража учетных данных пользователей путем эксплуатации SQL-инъекции в функционале поиска новостей в ленте

Модель угроз

Есть разные методологии, такие как:

- STRIDE
- DREAD
- PASTA
- и множество других

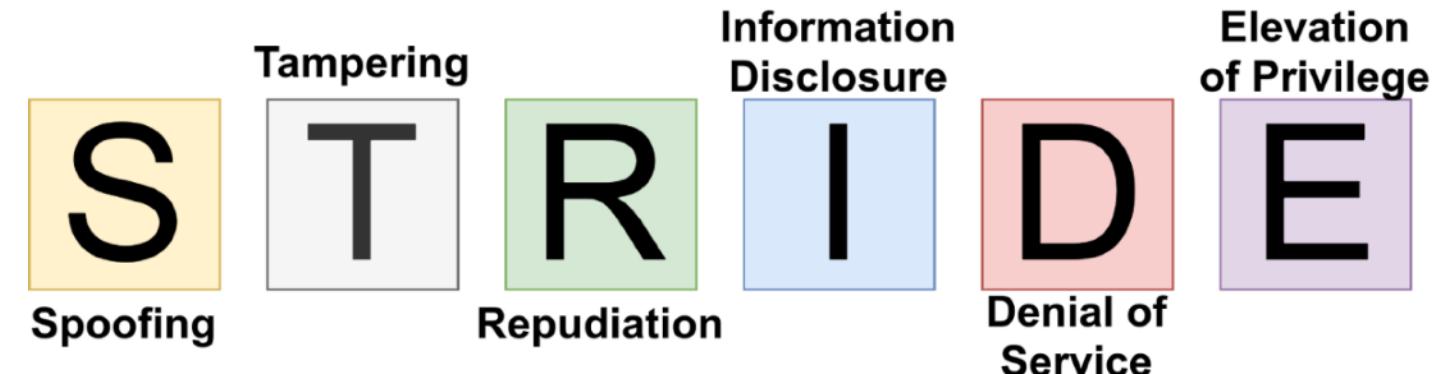


STRIDE

Состоит из следующих этапов:

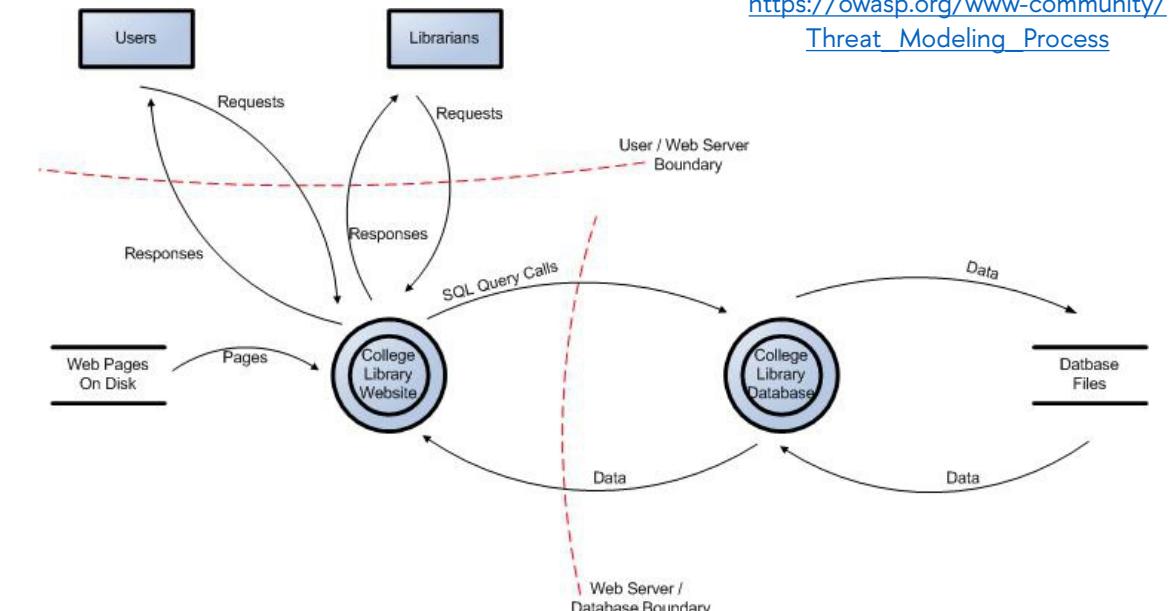
1. Определение области работ

- Общая информация
- Определение внешних зависимостей
- Определение входных/выходных точек
- Определение активов (Assets)
- Определение уровней доверия
- Построение диаграмм потока данных (DFD)



2. Определение угроз

3. Определение контр-мер и митигации
4. Оценка работы



Data Flow Diagram (DFD)

[https://owasp.org/www-community/
Threat_Modeling_Process](https://owasp.org/www-community/Threat_Modeling_Process)



Оценка уязвимостей

Качественный подход

- Критичная критичность
- Высокая критичность
- Средняя критичность
- Низкая критичность

Определяется методом экспертной оценки или модификацией количественного метода с учетом обстоятельств

Применяется, когда оценка количественным методом невозможна или имеются дополнительные обстоятельства, мешающие количественной оценке

Количественный подход

- CVSS** — The Common Vulnerability Scoring System
- Имеет несколько версий
 - Последняя — CVSS 4.0
 - Наиболее популярная — CVSS 3.1

EPSS — Exploit Prediction Scoring System

- Новое явление в мире наступательной безопасности
- Просчитывается и поддерживается организацией FIRST



CVSS 3.1

Rating	CVSS Score
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Base Score

7.5 (High)

Attack Vector (AV)
Network (N) Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC)
Low (L) High (H)

Privileges Required (PR)
None (N) Low (L) High (H)

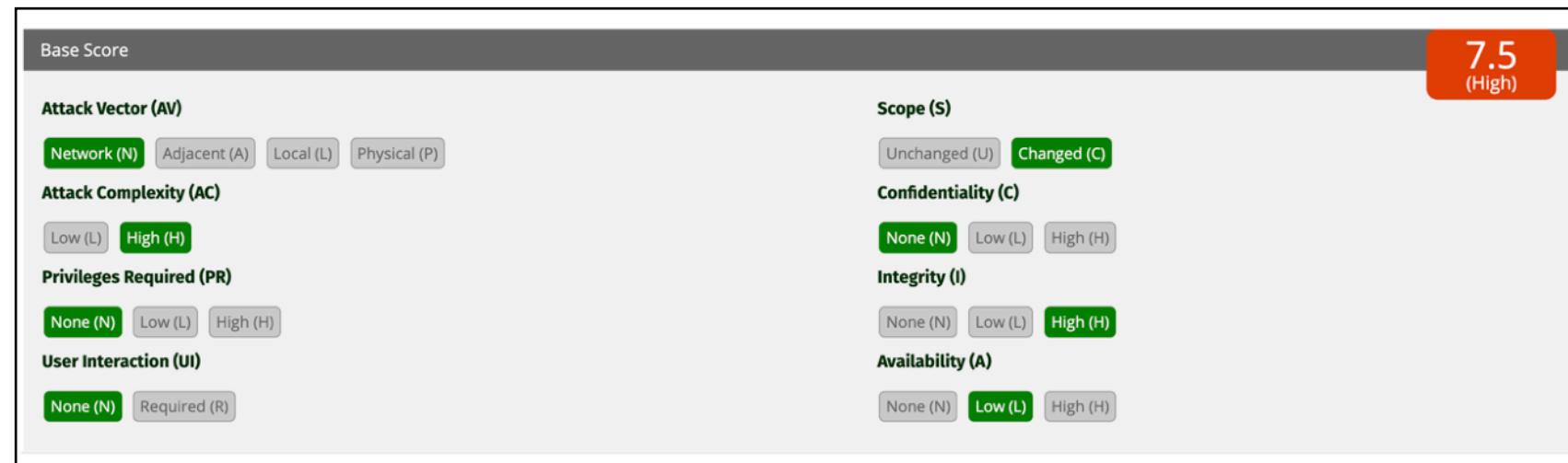
User Interaction (UI)
None (N) Required (R)

Scope (S)
Unchanged (U) Changed (C)

Confidentiality (C)
None (N) Low (L) High (H)

Integrity (I)
None (N) Low (L) High (H)

Availability (A)
None (N) Low (L) High (H)



«Расчет» уязвимости

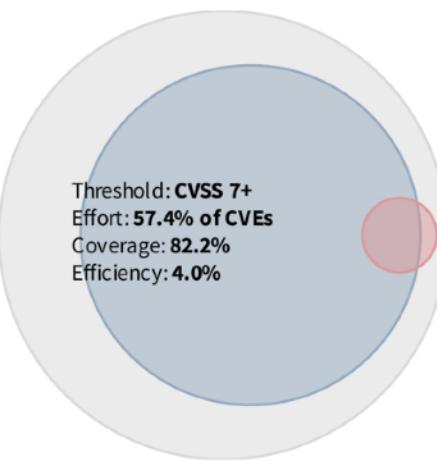
Определение критичности



EPSS

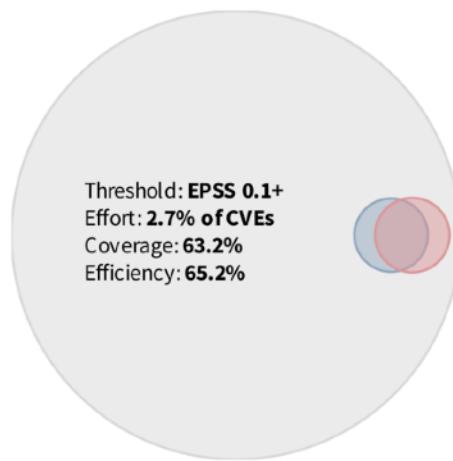
Comparing Metrics: CVSS 7+ vs EPSS 10%+

Pulling EPSS and CVSS scores from October 1st, 2023 and measuring predictive performance at arbitrary thresholds against exploitation activity October 1-30, 2023. Data is limited to CVEs with CVSS 3.x scores published in NVD as of Oct 1, 2023.



Source: <https://first.org/epss/model>

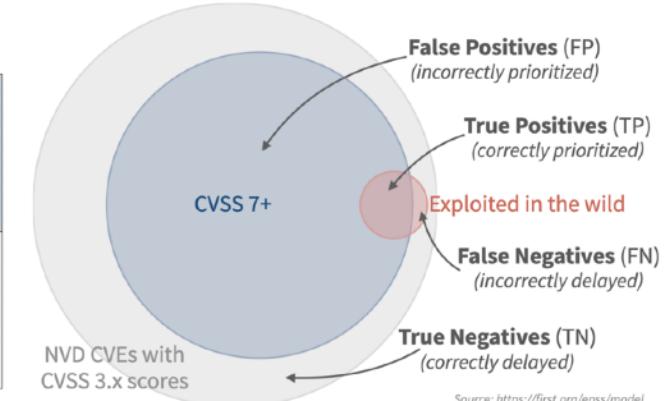
Threshold: EPSS 0.1+
Effort: 2.7% of CVEs
Coverage: 63.2%
Efficiency: 65.2%



Performance: Remediating CVSS 7 and above

Looking at the performance of CVSS scores produced October 1st, 2023, comparing against the observed exploitation activity recorded from Oct 1st to Oct 30th, 2023. CVSS threshold is (arbitrarily) set at 7.

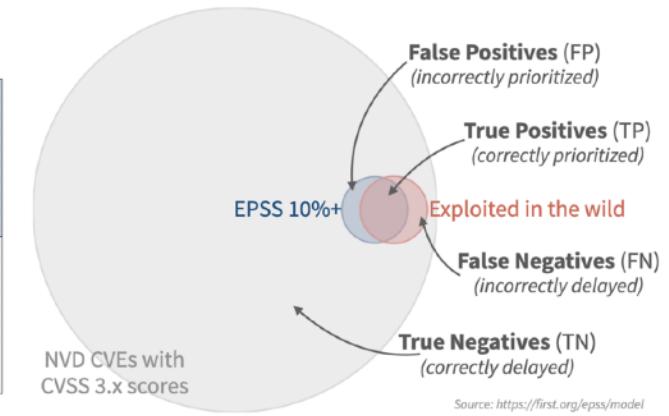
Our Decision...	Exploitation Activity...	
	Observed	Not Observed
Remediate (CVSS 7+)	3,166 (2.3%) True Positives (TP)	76,858 (55.1%) False Positives (FP)
Delay (< CVSS 7)	686 (0.5%) False Negatives (FN)	58,763 (42.1%) True Negatives (TN)



Performance: Remediating EPSS 10% and above

Looking at the performance of EPSS scores produced October 1st, 2023, comparing against the observed exploitation activity recorded from Oct 1st to Oct 30th, 2023. EPSS threshold is (arbitrarily) set at 10%.

Our Decision...	Exploitation Activity...	
	Observed	Not Observed
Remediate (EPSS 10%+)	2,435 (1.8%) True Positives (TP)	1,300 (0.9%) False Positives (FP)
Delay (< EPSS 10%)	1,417 (1%) False Negatives (FN)	134,321 (96.3%) True Negatives (TN)





CVE

CVE (Common Vulnerabilities and Exposures) - система идентификации и нумерации уязвимостей

The screenshot shows the CVE homepage (<https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2021-44228>). At the top, there are links for 'CVE List', 'CNAs', 'WG', 'Board', 'About', and 'News'. A banner at the top states 'TOTAL CVE Records: 240830'. Below it, two notices are displayed: one about the transition to the new CVE website (WWW.CVE.ORG) and another about the support for legacy CVE download formats ending on June 30, 2024. The main content area shows the details for CVE-2021-44228, including its description, references, and metrics.

<https://cve.mitre.org/>

The screenshot shows the NVD Detail page for CVE-2021-44228 (<https://nvd.nist.gov/vuln/detail/cve-2021-44228>). The page includes sections for 'VULNERABILITIES', 'CVE-2021-44228 Detail' (with a 'MODIFIED' status), 'Description' (describing the vulnerability in Apache Log4j), 'Metrics' (CVSS Version 4.0, CVSS Version 3.x, CVSS Version 2.0), and 'References to Advisories, Solutions, and Tools'. The right sidebar contains 'QUICK INFO' with details like the CVE Dictionary Entry, Published Date, Last Modified, and Source.

<https://nvd.nist.gov/>



CWE

CWE (Common Weakness Enumeration) - общий перечень проблем (слабостей) безопасности

<https://cwe.mitre.org/>

The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

Cross-site scripting (XSS) vulnerabilities occur when:

1. Untrusted data enters a web application, typically from a web request.
2. The web application dynamically generates a web page that contains this untrusted data.
3. During page generation, the application does not prevent the data from containing content that is executable by a web browser, such as JavaScript, HTML tags, HTML attributes, mouse events, Flash, ActiveX, etc.
4. A victim visits the generated web page through a web browser, which contains malicious script that was injected using the untrusted data.
5. Since the script comes from a web page that was sent by the web server, the victim's web browser executes the malicious script in the context of the web server's domain.
6. This effectively violates the intention of the web browser's same-origin policy, which states that scripts in one domain should not be able to access resources or run code in a different domain.

There are three main kinds of XSS:

- **Type 1: Reflected XSS (or Non-Persistent)** - The server reads data directly from the HTTP request and reflects it back in the HTTP response. Reflected XSS exploits occur when an attacker causes a victim to supply dangerous content to a vulnerable web application, which is then reflected back to the victim and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or e-mailed directly to the victim. URLs constructed in this manner constitute the core of many phishing schemes, whereby an attacker convinces a victim to visit a URL that refers to a vulnerable site. After the site reflects the attacker's content back to the victim, the content is executed by the victim's browser.
- **Type 2: Stored XSS (or Persistent)** - The application stores dangerous data in a database, message forum, visitor log, or other trusted data store. At a later time, the dangerous data is subsequently read back into the application and included in dynamic content. From an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. If one of these users executes malicious content, the attacker may be able to perform privileged operations on behalf of the user or gain access to sensitive data belonging to the user. For example, the attacker might inject XSS into a log message, which might not be handled properly when an administrator views the logs.
- **Type 0: DOM-Based XSS** - In DOM-based XSS, the client performs the injection of XSS into the page; in the other types, the server performs the injection. DOM-based XSS generally involves server-controlled, trusted script that is sent to the client, such as Javascript that performs sanity checks on a form before the user submits it. If the server-supplied script processes user-supplied data and then injects it back into the web page (such as with dynamic HTML), then DOM-based XSS is possible.



CWE Top 25

The screenshot shows the '2024 CWE Top 25 Most Dangerous Software Weaknesses' page. At the top, there are navigation links for Home, About, CWE List, Mapping, Top-N Lists, Community, News, and Search. A 'Top 25' badge is visible in the top right corner. The main content area lists the top 25 weaknesses with their ranks, names, and descriptions:

Rank	Name	Description
1	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	CWE-79 CVEs in KEV: 3 Rank Last Year: 2 (up 1) ▲
2	Out-of-bounds Write	CWE-787 CVEs in KEV: 18 Rank Last Year: 1 (down 1) ▼
3	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	CWE-89 CVEs in KEV: 4 Rank Last Year: 3
4	Cross-Site Request Forgery (CSRF)	CWE-352 CVEs in KEV: 0 Rank Last Year: 9 (up 5) ▲
5	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	CWE-22 CVEs in KEV: 4 Rank Last Year: 8 (up 3) ▲
6	Out-of-bounds Read	CWE-125 CVEs in KEV: 3 Rank Last Year: 7 (up 1) ▲
7	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	CWE-78 CVEs in KEV: 5 Rank Last Year: 5 (down 2) ▼
8	Use After Free	

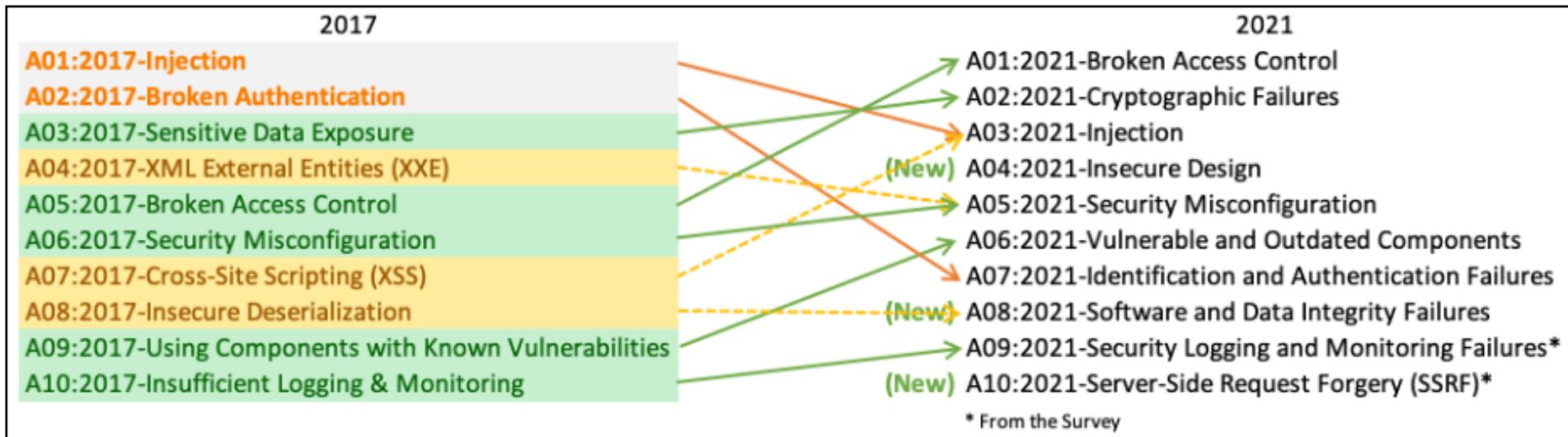
https://cwe.mitre.org/top25/archive/2024/2024_cwe_top25.html

The CWE Top 25		
Rank	ID	Name
1	CWE-787 ⚡	Out-of-bounds Write
2	CWE-79 ⚡	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
3	CWE-89 ⚡	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
4	CWE-416 ⚡	Use After Free
5	CWE-78 ⚡	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
6	CWE-20 ⚡	Improper Input Validation
7	CWE-125 ⚡	Out-of-bounds Read
8	CWE-22 ⚡	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
9	CWE-352 ⚡	Cross-Site Request Forgery (CSRF)

<https://www.sans.org/top25-software-errors/>



OWASP Top 10



<https://owasp.org/www-project-top-ten/>

- «Библия» пентестера



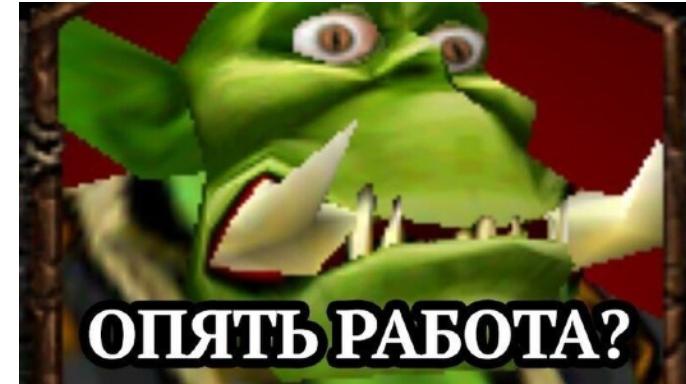


Виды работ

Виды работ

По характеру

1. Анализ защищенности
2. Внешний пентест
3. Внутренний пентест
4. Социотехническое тестирование
5. Физическое тестирование
6. Red Teaming
7. Purple Teaming
8. Research

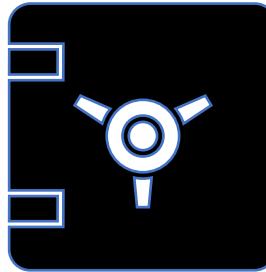


По типу

1. Черный ящик (Black Box)
2. Серый ящик (Grey Box)
3. Белый ящик (White Box)

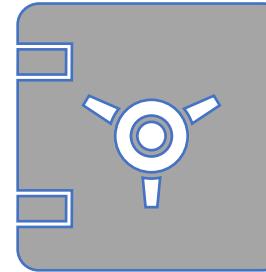


Виды работ (по типу)



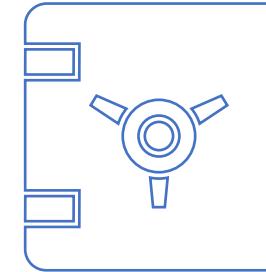
Черный ящик

1. **Нет знаний** об объекте тестирования



Серый ящик

1. **Ограниченные** знания об объекте тестирования
2. Имеются **учетные записи**
3. Имеется документация



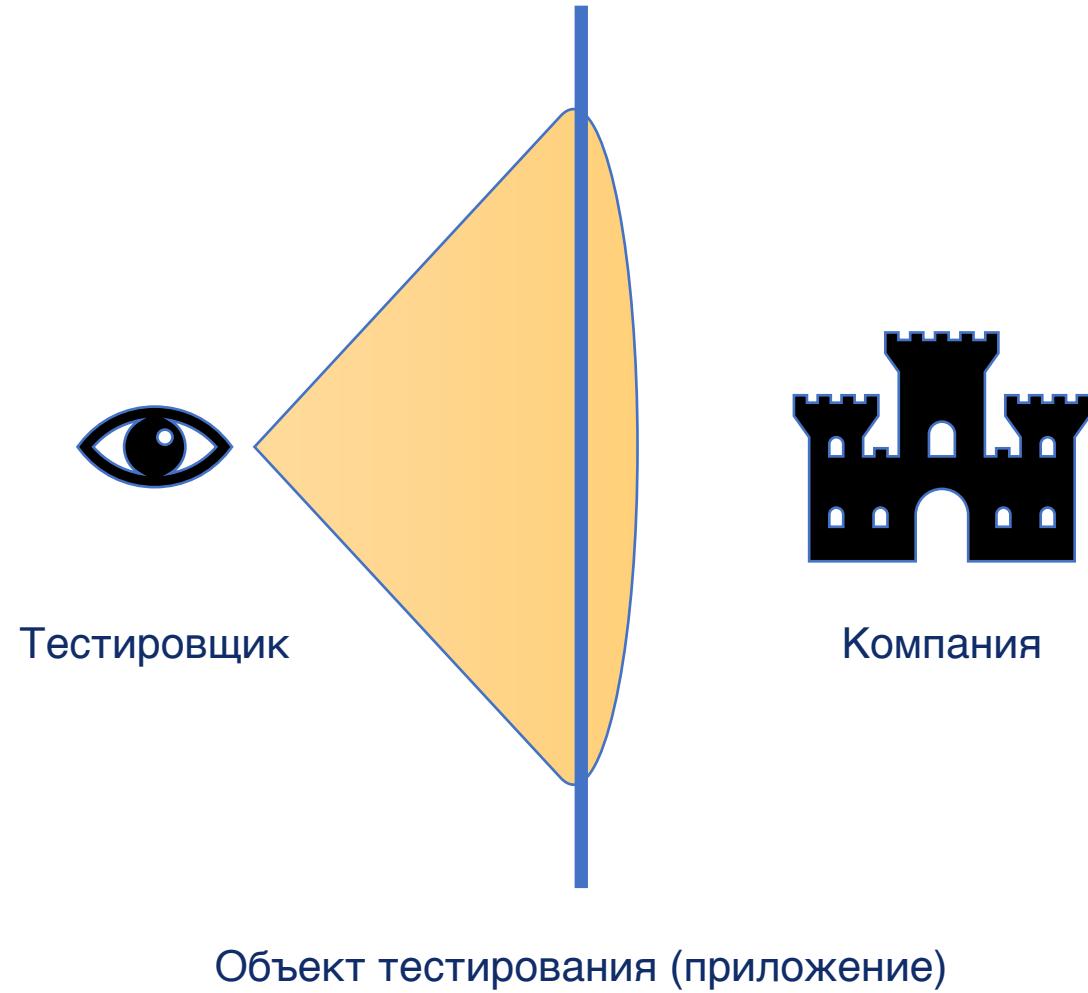
Белый ящик

1. **Полные** знания об объекте тестирования
2. Имеются учетные записи
3. Имеется документация
4. Имеется **исходный код**

Анализ защищённости

Цель: нахождение наибольшего количества уязвимостей

- Аналог «поиска в ширину»
- Делается акцент на количестве уязвимостей
- Имеют значения уязвимости любой критичности
- Объект тестирования может быть разным (веб-приложение, мобильное приложение, аппаратное средство и т.п.)
- Длится 1-2 недели

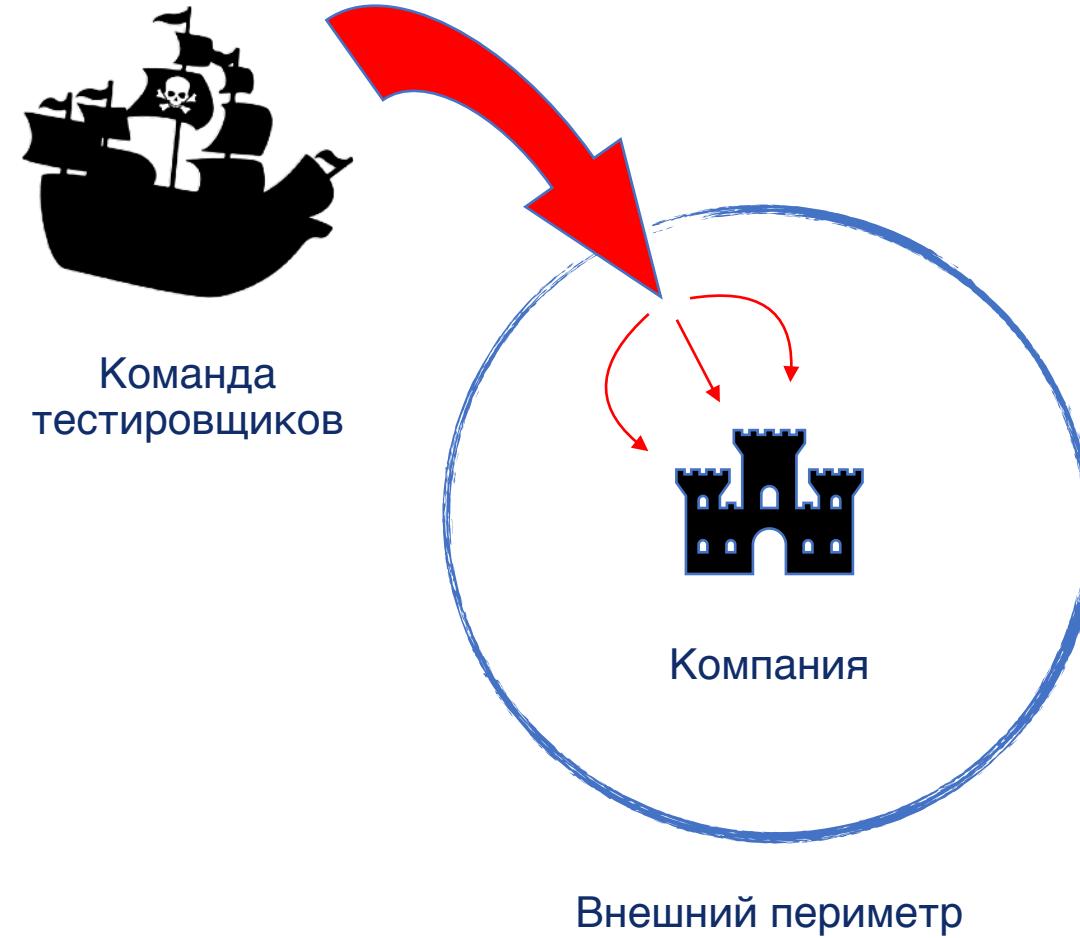




Внешний пентест

Цель: нахождение наиболее критичных уязвимостей, находясь снаружи периметра безопасности

- Аналог «поиска в длину»
- Делается акцент на критичности уязвимостей
- Одна из ключевых задач - «пробитие» периметра компании
- Проводится из внешней сети
- «Шумное» тестирование
- Длится 1-2 недели



Внутренний пентест

Цель: нахождение наиболее критичных уязвимостей, находясь
внутри периметра безопасности

- Аналог «поиска в длину»
- Делается акцент на критичности уязвимостей
- Одна из ключевых задач - реализация критических событий, влияющих на всю компанию
- Проводится из внутренней сети
- «Шумное» тестирование
- Длится 1-2 недели

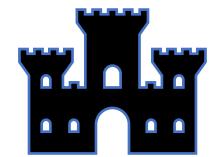
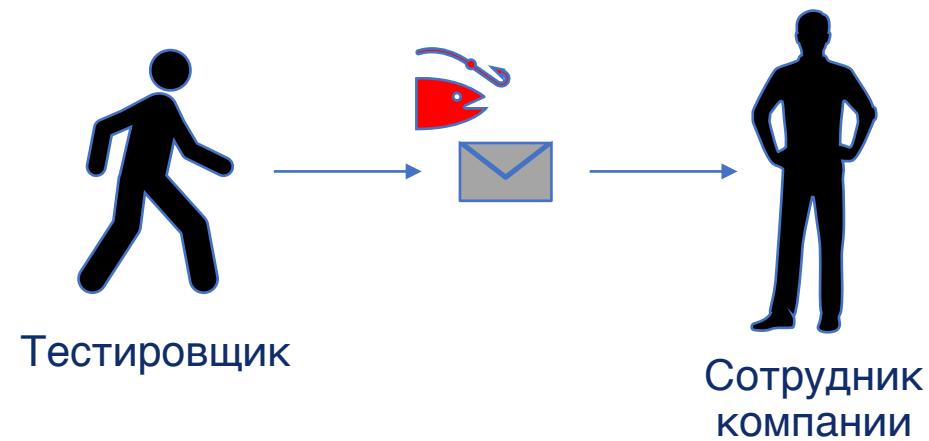




Социотехническое тестирование

Цель: нахождение слабых мест в сотрудниках компании

- Фишинг и прочие виды социальной инженерии
- Длится 1-3 недели



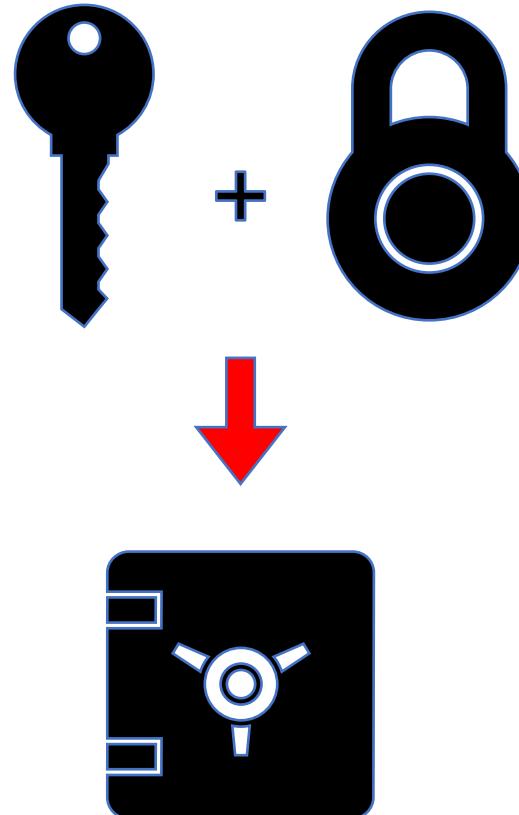
Компания



Физическое тестирование

Цель: нахождение слабых мест в физической безопасности
компании

- Локпикинг, карты местности, переодевание, аппаратные средства, сетевое оборудование, камеры, социальная инженерия и прочее
- Длится 1-4 недели



Red Teaming

Цель: имитация реальной атаки с мотивацией нанесения максимального ущерба и обхода средств защиты

- Включает в себя все виды тестирований
- Проводится с «боевыми» защитными механизмами компании без их уведомления
- Требует высокой аккуратности, точности и экспертизы
- Имитирует действия реальной целевой атаки (АРТ)
- «Тихое» тестирование
- Длится 1-3 месяца

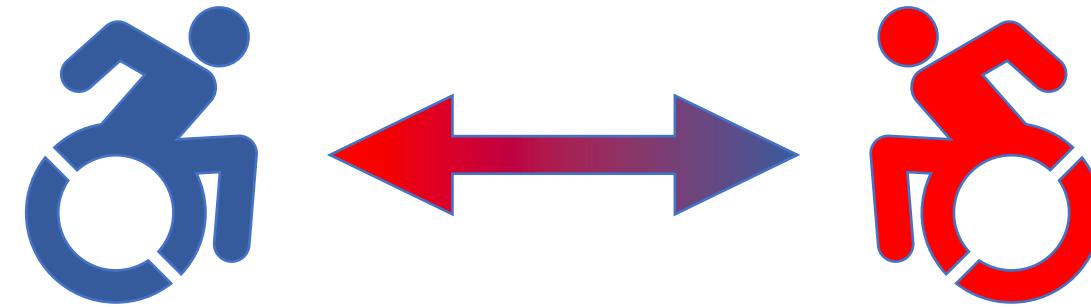




Purple Teaming

Цель: взаимное обучение и совершенствование
атакующей и защищающей команд

- Проводится внутри компании по договоренности команд
- «Красные» получают навыки атаки и обхода средств защиты
- «Синие» получают навыки детектирования угроз и имеют возможность улучшить средства обнаружения и защиты

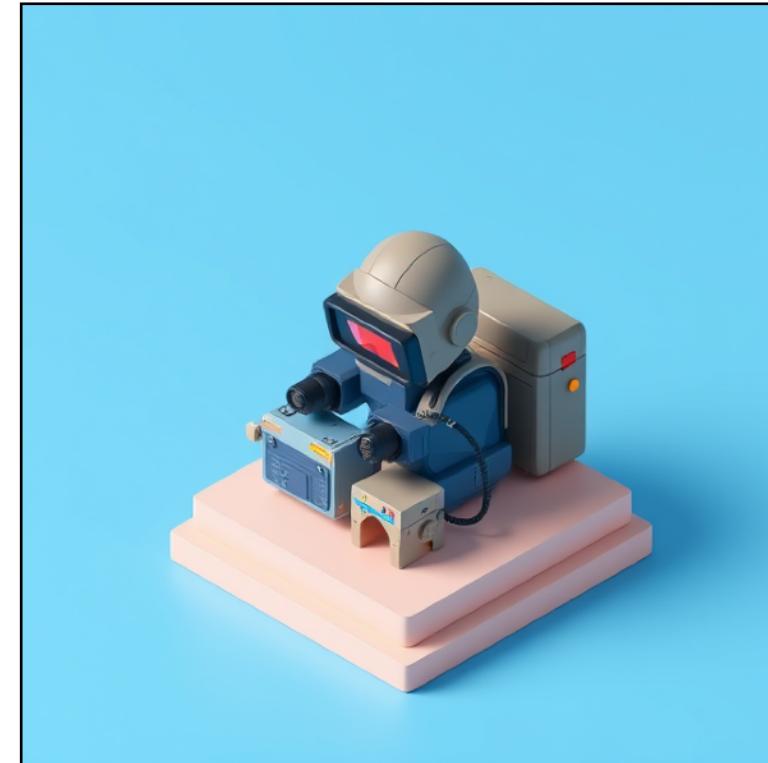




Research

Цель: обнаружение ранее неизвестных (0-day)
уязвимостей и написание эксплоитов к недавно
обнаруженным (1-day)

- Нужны для приобретения новых знаний и технологий
- Требует высокой экспертизы и знания узких технологий
- Поднимают имидж компании
- Помогают своим коллегам-пентестерам в сложных вопросах





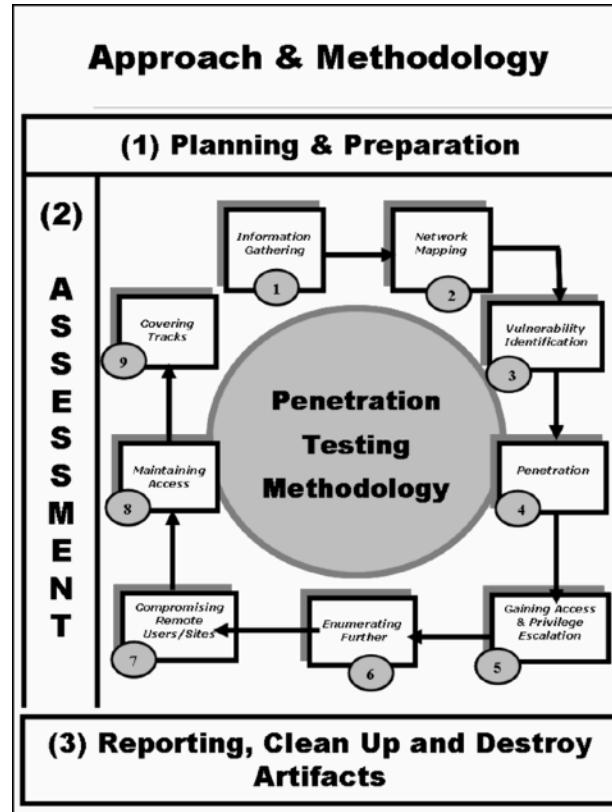
Этапы проведения тестирования на проникновение





Методологии проведения тестирования на проникновение

Методологии проведения тестирования на проникновение



ISSAF (Information systems
security assessment framework)

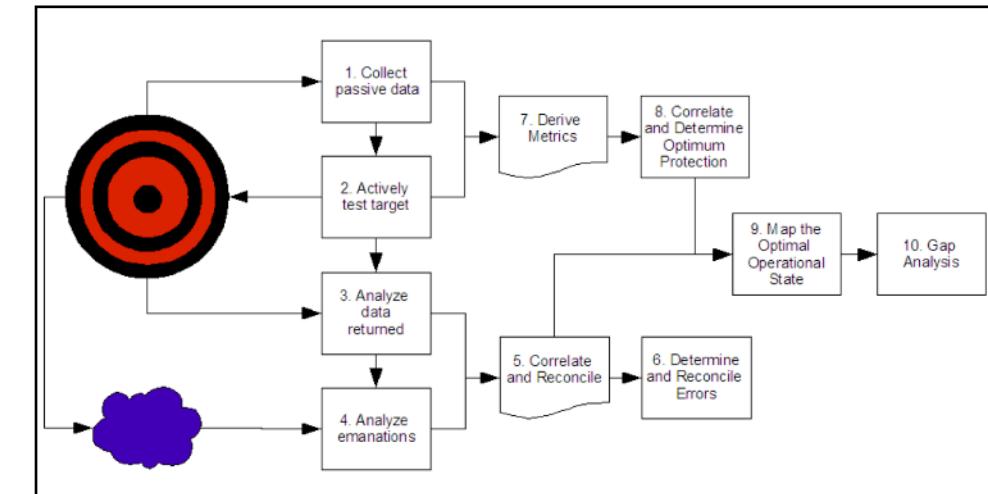
<https://untrustednetwork.net/files/issaf0.2.1.pdf>

Following are the main sections defined by the standard as the basis for penetration testing execution:

- Pre-engagement Interactions
- Intelligence Gathering
- Threat Modeling
- Vulnerability Analysis
- Exploitation
- Post Exploitation
- Reporting

PTES (Penetration Testing Execution Standard)

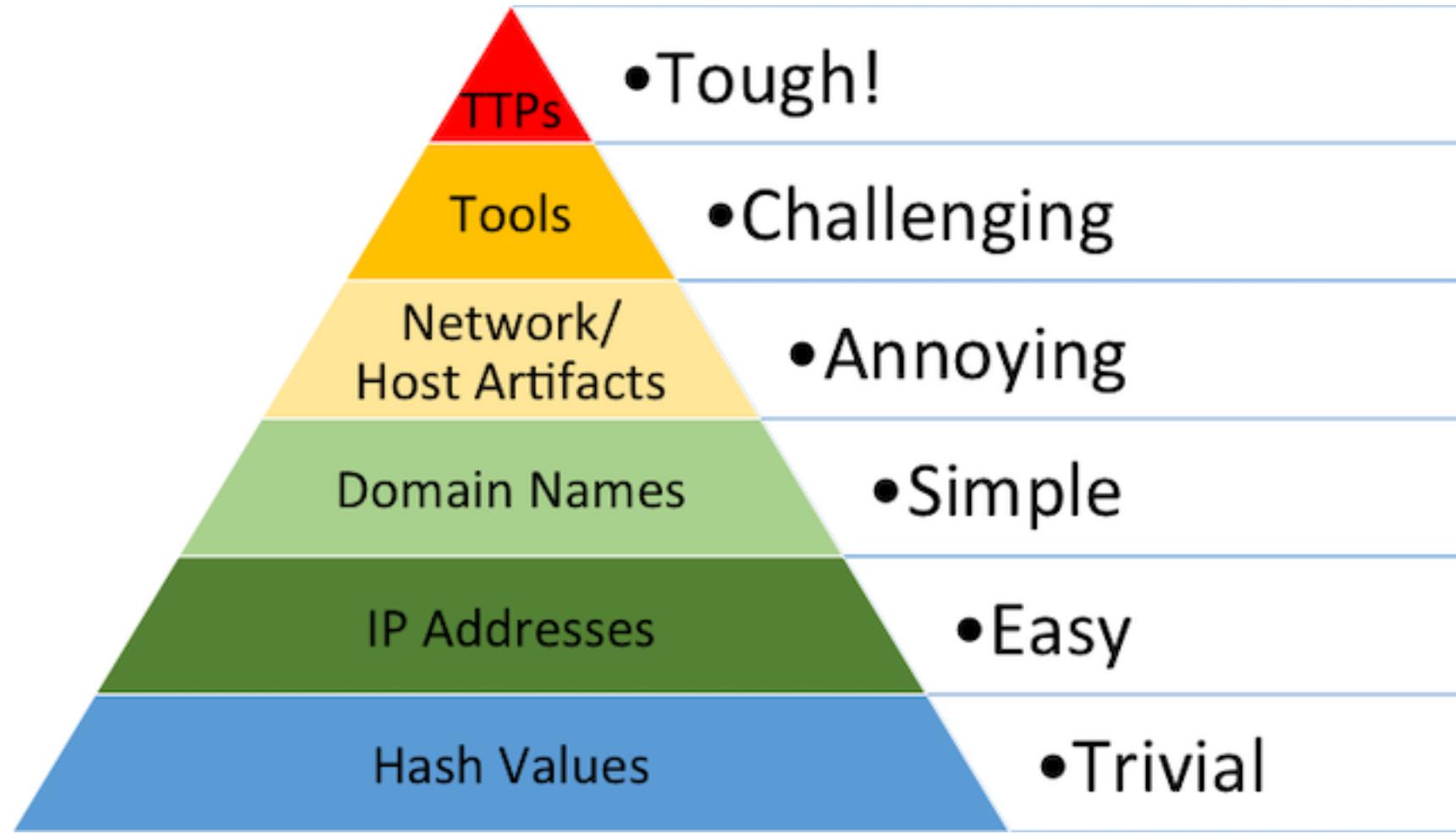
http://www.pentest-standard.org/index.php/Main_Page



OSSTMM (The Open Source
Security Testing Methodology
Manual)

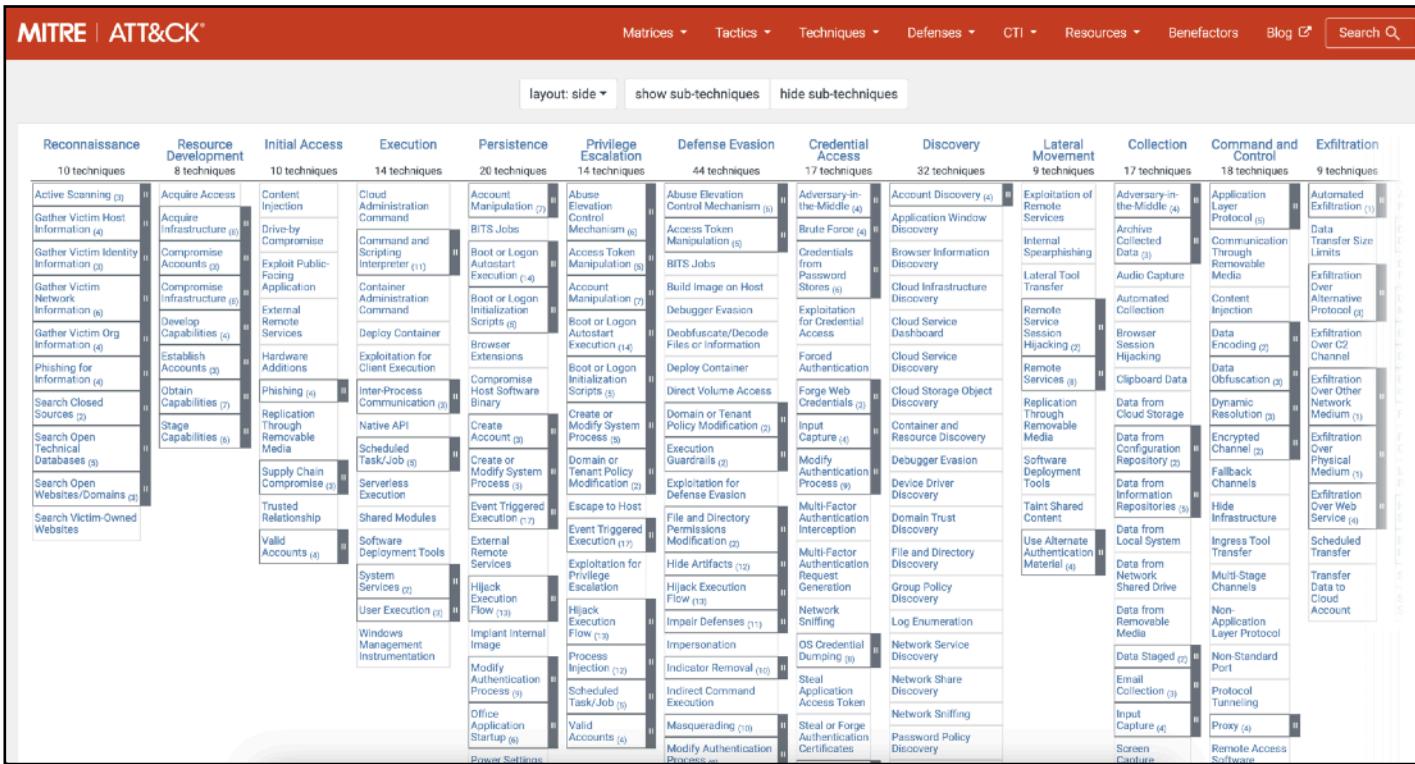
<https://www.isecom.org/OSSTMM.3.pdf>

Пирамида боли Дэвида Бьянко

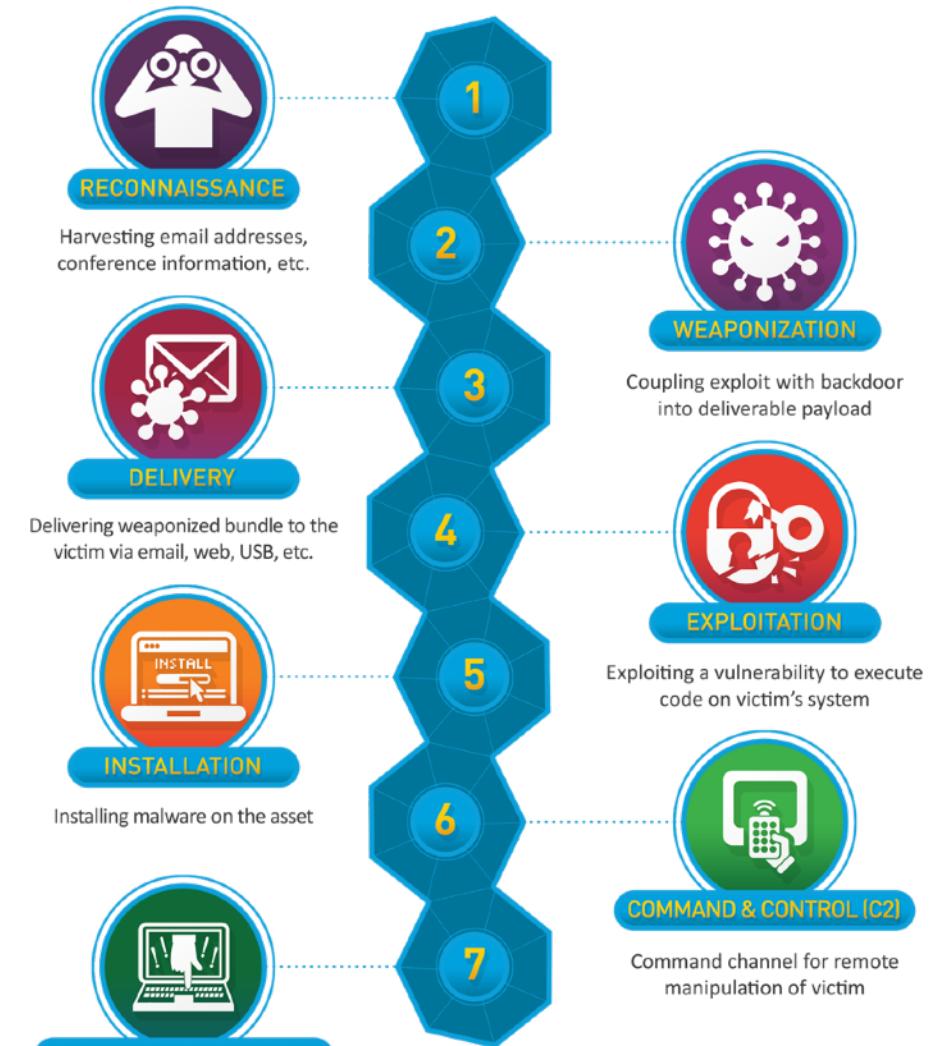




Mitre + Kill Chain



<https://attack.mitre.org/>



<https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>



Кафедра информационной
безопасности киберфизических
систем

Лекция 1: Введение в наступательную
безопасность

Методологии проведения
тестирования на проникновение

46



@LEXA_MALOSPAAL



@HUN7_OR_B3_HUN73
D

